UBI-CA : A Clustering Algorithm for Ubiquitous Environments

Rim Helali
Higher Institute of Management of Tunis
Research Laboratory SOIE
Tunis, Tunisia
rim.helali@gmail.com

Nadia Ben Azzouna
National School of Computer Science
Research Laboratory SOIE
Manouba, Tunisia
nadia.benazzouna@ensi.rnu.tn

Khaled Ghedira
Higher Institute of Management of Tunis
Research Laboratory SOIE
Tunis, Tunisia
khaled.ghedira@isg.rnu.tn

***Abstract*—This paper describes a new Clustering Algorithm for UBIquitous environments (UBI-CA). These types of environments are dynamic in nature due to the mobility of nodes. The constant motion of devices and their unexpected appearance or disappearance could perturb the stability of the network topology. Clustered architectures seem to be adequate to address such a challenge. In the proposed algorithm, inspired by the Weighted Clustering Algorithm (WCA), we aim for the reduction of the computation and communication costs by electing the most suited cluster heads on the base of a calculated weight. We propose to emphasize on the mobility and the number of neighboring nodes while calculating its weight in order to cope with ubiquitous environments specificities. The cluster structure is maintained dynamically as nodes move by defining a nodes dynamicity management method. Simulation results prove that the proposed algorithm ensure a good scalability for our ubiquitous system as the number of nodes increases. Furthermore, when compared with the original algorithm WCA, UBI-CA shows a better stability with increasing transmission range of nodes.**

Keywords-ubiquitous environments; clustered architecture; mobility; scalability.

I. INTRODUCTION

Ubiquitous computing aims to exchange and share services anywhere, anytime. In addition to the abundant number of users, ubiquitous environments are characterized by the heterogeneity of devices (computers, mobile phones, personal digital assistants, etc.) and their dynamicity (the mobility of devices in the environment and their unexpected appearance or disappearance [1]). A distributed architecture is well adapted for this type of networks in order to prevent the problems of bottleneck and Single Point Of Failure (SPOF) caused by the big number of connected devices. Furthermore, the mobility of nodes should be considered in the deployed architecture. Clustered architectures are proven to be adequate for ubiquitous environments thanks to their distributed deployments and to their ability to be robust in the face of topological changes caused by nodes motion [2]. In that context, a clustered architecture for ubiquitous environments should be able to dynamically adapt itself with the changing network configurations [3] and to reduce the communication costs. Therefore, a clustering algorithm UBI-CA is proposed in this paper allowing the division of the geographical region into small zones [3] in order to reduce the cost of searching services. Hence, we aim in this work to define a clustering algorithm that manages the high dynamicity of the environment while guaranteeing the scalability and the stability requirements [4].

The rest of this paper is organized as follows. Section 2 addresses related work and identifies some limitations. Section 3 describes the proposed algorithm. Section 4 presents simulation results. Finally, Section 5 concludes the paper and discusses some future directions.

II. RELATED WORK

Several clustering algorithms have been proposed in the literature especially for ad hoc networks. According to Agarwal et al. in their review of clustering algorithms [5], clustering algorithms for mobile ad hoc networks (MANET) could be classified in two categories, single metric based clustering and multiple metrics based clustering. In single metric based clustering, only one performance factor is considered to select cluster heads. A number of clustering algorithms for this category has been proposed in the literature. In Lowest ID cluster algorithm (LIC) [6], the node with the lowest id is chosen as a cluster head. In Highest connectivity clustering algorithm (HCC) [7], each node broadcasts its id to the nodes that are within its transmission range. The node with the highest number of neighbors is elected as cluster head. In K-CONID [8], two methods are considered in order to choose cluster heads. Connectivity is used as a first criterion and lower ID as a secondary criterion. The above mentioned algorithms suffer from the problem of cluster head overloading since each algorithm deals with only one parameter [3]. To solve this problem, multiple metrics based clustering has been proposed. In this category, a number of metrics, such as node degree, residual energy capacity and, moving speed are taken into account [5]. In this context, Basagni proposed two clustering algorithms [9], distributed clustering algorithm (DCA) and distributed mobility adaptive clustering algorithm (DMAC). In DCA, a node is chosen to be a cluster head if its weight is higher than any of its neighbors weight; otherwise, it joins a neighboring cluster head. The weight is generic and is calculated according to nodes mobility related parameters [5]. It is assumed in DCA that the network topology does not change during the execution of the algorithm. The DMAC algorithm was proposed as an extension to DCA allowing the adaptation of the algorithm to the network topology. In Weighted Clustering Algorithm (WCA) [3], the authors focused mainly on the stability of

 ISBN: 978-1-61208-330-8

the network topology. They take into consideration the ideal degree which refers to the ideal number of nodes a cluster head can handle, transmission power, mobility, and battery power of mobile nodes. The cluster head election procedure is invoked on-demand and aims to reduce the computation and communication costs. Nevertheless, WCA suffers from several drawbacks especially if applied in ubiquitous environments. For example, using Global Positioning System (GPS) in order to calculate nodes mobility appears to be not adequate due to devices high heterogeneity. Furthermore, considering the cumulative time during which a node acts as a cluster head to calculate the battery power is of a low relevance in our case because knowing the consumed battery power of a node does not reflect the time during which a node can play the role of a cluster head after being elected.

Thus, in our proposed algorithm, we adapt the Weighted Clustering Algorithm (WCA) originally proposed for ad hoc mobile networks to meet ubiquitous environments challenges.

III. PROPOSED ALGORITHM

In this section, we present our defined clustering algorithm UBI-CA which is inspired from the WCA algorithm. We justify the choice of WCA by the fact that this algorithm considers several parameters like transmission power, mobility and battery power of mobile nodes [3]. These parameters are very significant for the election of the cluster heads, especially in ubiquitous environments where nodes are characterized by their high dynamicity. In our proposed architecture for a semantic and dynamic cluster based web service discovery system presented in [10], each cluster is organized in a two level hierarchical architecture containing a Directory Server (DS) and a number of web service servers (WSS) [11]. Thus, we are dealing, in this algorithm, with three types of nodes :

- A Directory Server (DS): responsible of a certain region in the ubiquitous environment. It maintains descriptions of all existing web services distributed within this region [11].
- A backup Directory Server (backup DS): used to replace the DS in case of failure in order to guarantee service availability.
- A Web Service Server (WSS): containing a number of web services and their descriptions. A WSS is assigned to a DS in the ubiquitous environment. This DS becomes its gate to the entire ubiquitous environment [11].

The following subsection describes in details the UBI-CA algorithm in order to form the proposed architecture. A preliminary version of this algorithm appeared in [10].

A. DSs election and clusters formation

Initially, there are no elected DSs in the environment. The DSs list is generated for the first time by invoking the DSs election procedure at system activation time. The procedure begins by nodes broadcasting to their immediate neighbors (i.e., one-hop neighbors) their ID. Then, only the nodes with a transmission delay lower than a predefined threshold are chosen. The motivation behind this restriction is essentially

Algorithm 1 UBI-CA

Input : listID, n, TD, P_v, δ
Initialization : dv=0
for $i = 1$ to n **do**
  for $j = 1$ to $n-1$ **do**
    if Transmission Delay $\langle$ TD **then**
      $d_v = d_v + 1$
      $add(listID(j), N)$
    end if
  end for
  $\Delta_v = |d_v - \delta|$
  $M_v(t-\Delta t) = \frac{1}{|U|}\sum_{i,j\in U}\frac{M_{ij}(t-\Delta t)}{\Delta t}$
  $W_v = \frac{(1+b\Delta_V + cM_v)^{\alpha}}{P_v}$
end for

listId = list of the identifiers of one-hop neighbors, TD = a predefined transmission delay threshold that a node cannot exceed to be considered as a neighbor, n = the total number of one-hop neighbors, N = the set of one-hop neighbors of the node v with transmission delay $\langle$ TD, P_V = Battery power at the time t.

Figure 1: UBI-CA

the reduction of the distance between a node and its neighbors. The procedure, as illustrated in Fig. 1, consists of electing the DSs on the base of a calculated value W_v. In the original WCA algorithm, the value W_v calculated for every node is a sum of components with certain weighting factors chosen according to the system needs [3]. Due to the dynamicity of the ubiquitous environment and in order to guarantee the system stability, we judge that the nodes mobility M_v and the degree of nodes (i.e., the number of neighboring nodes D_v) are more relevant for DS stability than battery power in the DSs election procedure. Therefore, the value W_v is calculated as (1) :

$$W_v = \frac{(1+b\Delta v + cMv)^{\alpha}}{Pv} \quad (1)$$

After calculating its W_v, each node v sends its value to its neighbors. This exchange of messages permits the nodes to be aware of the node with the lowest W_v in their neighbors set which is elected as a DS. Non elected nodes (i.e., WSSs) send a membership query to their local DS. Thus, each new elected DS forms a list of its cluster members. If a node receives at least one membership query despite the fact that it has not the lowest W_v in its neighbors set (it has the lowest W_v in another node neighbors set), it has to disjoin the cluster to which it has been assigned as a WSS by sending a disjunction query to the local DS and should be elected as a DS. Each elected DS chooses from the cluster the node that has the second lowest W_v as a backup DS. If a node has an empty neighbors set due to transmission delay restriction, it is then elected as a DS in its region [10].

In the next subsections, we describe in details the three components forming the W_v value.

B. The degree difference Δ_v

The first component Δ_v computes the degree-difference for each node v by calculating the difference between the ideal number of nodes a DS can support δ and the number of

Algorithm 2 Nodes Mobility

Input : d_v, array x, array y
for $i - 1$ to d_v **do**
  for $j = i + 1$ to $d_v - 1$ **do**
    if $i \in N_j$ and $j \in N_i$ **then**
      $add(i, j, U)$
      $\phi_{vj} = cos^{-1} \frac{d_{vi}^2 + d_{ij}^2 + d_{vj}^2}{2 d_{vi} d_{ij}}$
      $l = \sqrt{d_{vi}^2 + (\frac{d_{ij}}{2})^2 - d_{vi} d_{ij} cos\phi_{vj}}$
      $\theta = cos^{-1} \frac{l^2 + (\frac{d_{ij}}{2})^2 - d_{vj}^2}{l d_{ij}}$
      $y'(i) = l cos\theta$
      $x'(i) = l sin\theta$
    end if
  end for
end for
for $i = 1$ to $|U|$ **do**
  $M_i = \sqrt{(y'^2 - y^2) + (x'^2 - x^2)}$
  $M_v = M_v + \frac{M_i}{\Delta t}$
end for
$Mv(t - \Delta t) = \frac{1}{|U|} M_v$
return M_v

Figure 2: Nodes mobility

neighbors d_v. It helps, thus, choosing the node with the nearest degree to the ideal degree δ as a DS. This is to ensure that the DSs are not overloaded and the efficiency of the system is maintained at the expected level [3].

C. Nodes mobility

The component M_v represents the mobility of nodes (i.e., nodes changing their location over time). To calculate nodes mobility, we choose, rather than using the mobility metric proposed in the original algorithm that is based on a localization system, to use the mobility metric proposed in [12] that does not depend on any location system (e.g., GPS) and can fully capture the relative motions between a node and its neighborhood, in real-time, using simple triangulation [12].

As illustrated in Fig. 2, the set U represents for each node v the pair of nodes that are both neighbors to the node v as well as one-hop neighbors to each other. Each node periodically measures the distances to its one-hop neighbors using the "transmission delay" between nodes (i.e., d_{vi}, d_{ij}, d_{vj}) . The measurements are sent periodically with a frequency of sending of $1/\Delta t$ [12]. l calculates the distance from node v to the midpoint of the line between nodes i and j as well as the included angle θ to interpret the relative position between node v and $pair(i, j)$ [12].

D. Battery power

The last component P_v, represents the battery power of a node. We choose in our algorithm rather than computing the consumed battery power used in WCA, to focus on the remaining battery power and to elect the node with the highest value. We assume that the remaining battery power is more relevant in our case due to the fact that a node may initially have limited battery power. So, knowing the consumed battery power of a node does not reflect the time during which a node can play the role of a DS after being elected.

E. Nodes dynamicity management

Due to nodes mobility caused by the dynamic nature of the ubiquitous environment, a node may be detached from its local DS and looks for a new DS; a reaffiliation is thus needed. New DSs could be added to the set if a node cannot find a DS to which it can be affiliated due to the overload of neighboring DSs. In addition to nodes mobility, unavailability of nodes (i.e., breakdown, battery power extinction) has to be managed especially for DSs and backup DSs. We discuss each of these cases in details in what follows [10].

- A node detecting a new DS with higher signal strength than its local DS: The node sends an affiliation request. The DS updates its list and informs the old DS after accepting the request.
- The arrival of a new node which sends an affiliation request to all its neighboring DSs but no response is received (due to the overloading of neighboring DSs): The DS election procedure is invoked, but only neighboring clusters are involved. After reforming the clusters, each old DS sends to its local DS its registered service descriptions, and each new DS sends a DS replacement query to all other DSs in the environment.
- The failure of a DS (If no messages are received by the backup DS during a specific period of time): The backup DS initiates the DS election procedure by sending a DS election query to the WSSs members of the cluster. After electing the DS, the backup DS sends its registered service descriptions to the new DS which chooses the node with the second lowest W_v as its new backup DS and sends to the other DSs in the environment a message informing them of its new status.
- The failure of a backup DS (if no acknowledgements are received during a specific period of time) : The DS selects the node that has the smallest W_v from its WSSs to be the new backup DS (Nodes have to recalculate their W_v)
- The failure of both DS and backup DS (the unavailability of the DS is detected by the WSSs): The WSSs have to elect a new DS among them by recalculating their W_v and choosing the WSS with the smallest W_v. The second WSS to have the smallest W_v is chosen as a backup DS.

IV. Simulation Study

In order to study the performance of our algorithm UBI-CA, a simulator was developed using omnet++, an extensible, modular, component-based C++ simulation library and framework. OMNeT++ provides a component architecture for models. Components (modules) are programmed in C++, and then assembled into larger components and models using a high-level language (NED) [13]. For the simulation study, the following aspects are observed: (i) the scalability of the system, using UBI-CA, in terms of maintaining a good performance as the number of nodes increases, (ii) the stability of the proposed algorithm in terms of DSs number while the number of nodes and transmission range increase.

 ISBN: 978-1-61208-330-8

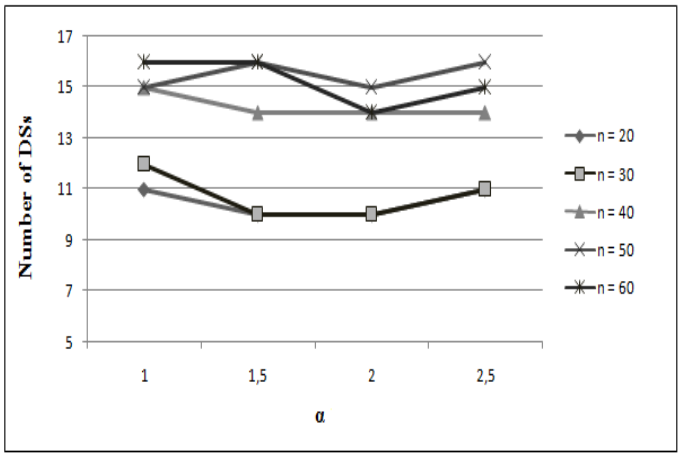

Figure 3: Number of DSs variation with α (transmission range =100)

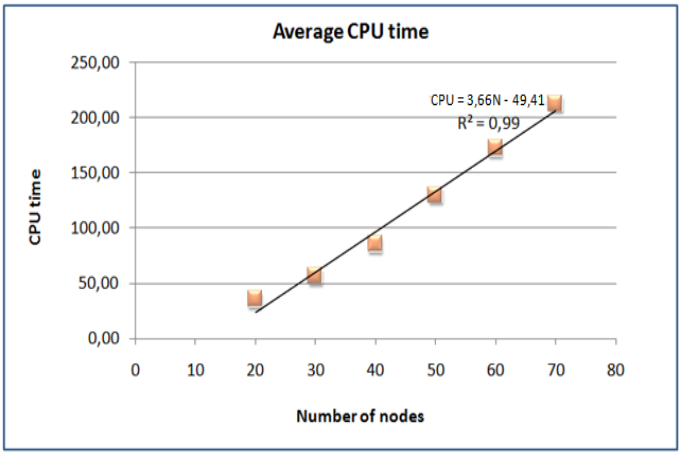

Figure 4: Average CPU time variation

A. Simulation Environment

The experiments were all conducted using an Intel (R) Pentium (R) 2.00 GHz computer with 3 Go RAM, running Windows 7. We simulate a system of N nodes on a 900 x 900 grid. In our simulation experiments, N was varied between 20 and 70, and the transmission range was varied between 10 and 100. The values used for the simulation are c = 0.5 and b = 0.5, and for the ideal degree δ =10 . Note that these values are arbitrary and should be adjusted according to the system requirements. The value of α (used in (1)) is set to be equal to 2. Note that this value is obtained by varying the parameter between 1 and 2.5 and, as shown in Fig. 3, α = 2 provides the lowest number of DSs.

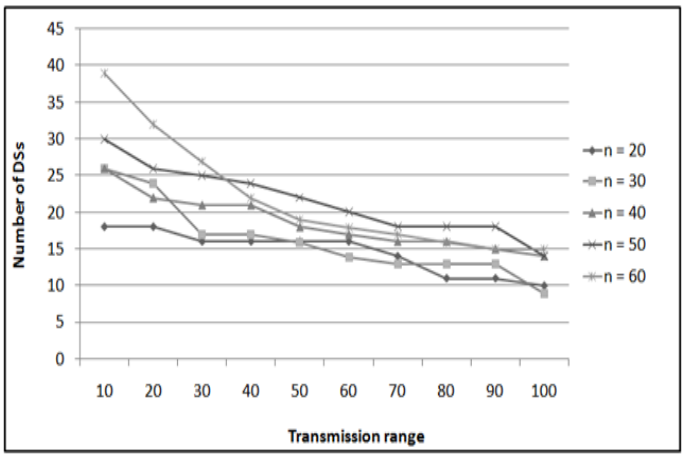

Figure 5: Number of DSs variation

B. Experimental results

In order to analyze how our system scales with increasing number of nodes, we varied the transmission range between 10 and 80 meters and measured the variation of the average CPU time with respect to the number of nodes N. We observe, as shown in Fig. 4, that the increase of the average CPU time is linear. The linearity is proven by the correlation coefficient R^2 value (calculated according to PEARSON function [14]) which is close to 1 indicating an excellent linear fit. This shows that our system has a good performance as the number of nodes increases.

The stability of our system is tested by measuring the variation of the number of DSs with respect to the transmission range. The results are shown for varying N. We observe, as noticed in Fig. 5, that the number of DSs decreases with the increase of the transmission range. This is due to the fact that a DS with a large transmission range will cover a larger area. The results show that the number of DSs is not influenced by the variation of the number of nodes which emphasizes the stability of the system.

Comparing our results with those found using WCA algorithm (as shown in Fig. 6), we notice that the UBI-CA results are proven to be better in terms of number of DSs. This shows that the emphasis put on both nodes mobility and degree difference while calculating the W_v value is confirmed to be adequate for ubiquitous environments.

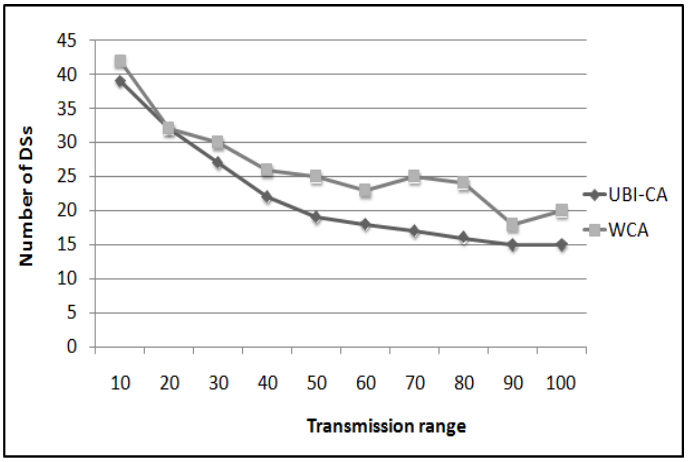

Figure 6: Number of DSs variation for UBI-CA and WCA (N = 60)

 ISBN: 978-1-61208-330-8

V. Conclusion and future work

In this paper, we introduced a new clustering algorithm for ubiquitous environments named UBI-CA which is inspired from the original Weighted Clustering Algorithm (WCA). The choice of WCA is based on the fact that this algorithm takes into consideration several parameters like transmission range, mobility and battery power of mobile nodes. However, this algorithm suffers from several problems that we tried to resolve. Therefore, to calculate nodes mobility, we used the mobility metric proposed in [12] which is independent from any location system (e.g., GPS). Furthermore, battery power in our algorithm represents the remaining battery power of a node rather than the consumed power used in the original algorithm. In order to calculate the W_v value, we focused mainly on the nodes mobility M_v and the degree of nodes Δ_v. In order to evaluate our approach, we observed the scalability and the stability of the proposed algorithm. Our future work includes the extension of the proposed system to support quality of service during the web service selection in order to meet the challenges of ubiquitous environments such as invisibility.

References

[1] I. Abdennadher, "Une approche pour l'assurance des qualites de services des systemes publier/souscrire deployes sur un reseau mobile ad-hoc," Master's thesis, National Engineering School of Sfax, Tunisia, 2008.

[2] C. R. Lin and M. Gerla, "Adaptive clustering for mobile wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 15, pp. 1265–1275, 1997.

[3] M. Chatterjee, S. K. Das, and D. Turgut, "Wca: A weighted clustering algorithm for mobile ad hoc networks," *Journal of Cluster Computing (Special Issue on Mobile Ad hoc Networks)*, vol. 5, pp. 193–204, 2001.

[4] M. Satyanarayanan, "Pervasive computing: Vision and challenges," *IEEE Personal Communications*, vol. 8, pp. 10–17, 2001.

[5] R. Agarwal1, R. Gupta, and M. Motwani3, "Review of weighted clustering algorithms for mobile ad hoc networks," *GESJ: Computer Science and Telecommunications*, vol. 33, pp. 71–78, 2012.

[6] D. J. Baker and A. Ephremides, "A distributed algorithm for organizing mobile radio telecommunication networks." in *In proceeding of the 2nd International Conference on Distributed Computing Systems*, 1981, pp. 476–483.

[7] M. Gerla and J. T. chieh Tsai, "Multicluster, mobile, multimedia radio network," *Journal of Wireless Networks*, vol. 1, pp. 255–265, 1995.

[8] G. Chen, F. Nocetti, J. Gonzalez, and I. Stojmenovic, "Connectivity-based k-hop clustering in wireless networks," in *Proceedings of the 35th Annual Hawaii International Conference on System Sciences (HICSS'02)-Volume 7 - Volume 7*, ser. HICSS '02, 2002, pp. 2450–2459.

[9] S. Basagni, "Distributed clustering for ad hoc networks," in *International Symposium on Parallel Architectures, Algorithms, and Networks*, 1999, pp. 310–315.

[10] R. Helali, N. B. Azzouna, and K. Ghdira, "Towards a semantic and dynamic cluster based web service discovery system for ubiquitous environments." in *ICEIS (2)*. SciTePress, 2012, pp. 295–300.

[11] T. Xu, B. Ye, M. Kubo, A. Shinozaki, and S. Lu, "A gnutella inspired ubiquitous service discovery framework for pervasive computing environment," *8th IEEE International Conference on Computer and Information Technology, CIT*, pp. 712 – 717, 2008.

[12] Z. Li, L. Sun, and E. C. Ifeachor, "Gps-free mobility metrics for mobile ad hoc networks," *IET Communications*, vol. 5, pp. 1–18, 2007.

[13] "Omnet++ documentation," http://omnetpp.org/documentation, accessed: 16.12.2013.

[14] K. Pearson, "Notes on regression and inheritance in the case of two parents," 1985, pp. 240–242.

 ISBN: 978-1-61208-330-8